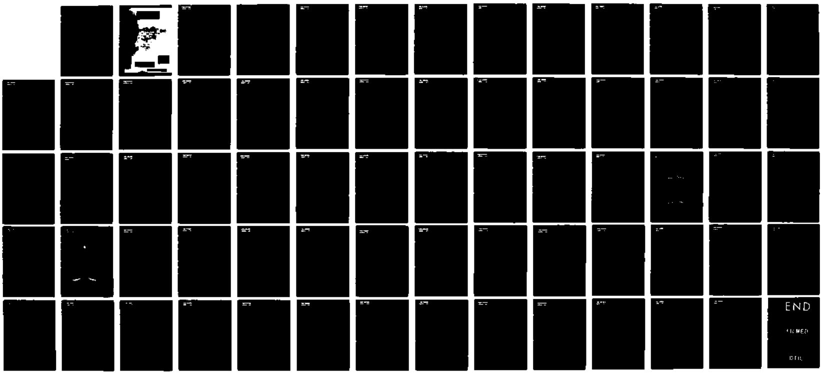
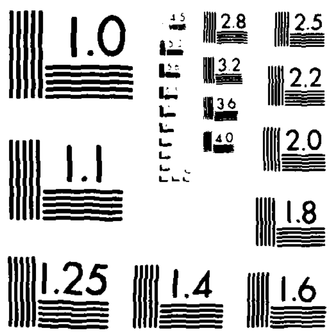
AD-A149 245 ACOUSTIC ENVIRONMENT SIMULATION STUDY; ACOUSTIC INTRUSION SENSOR PERFORMANCE(U) TIME SERIES ASSOCIATES PALO ALTO CA L ENOCHSON ET AL. JAN 83 N60921-82-M-6028 1/1

UNCLASSIFIED F/G 14/2 NL




1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

TIME SERIES ASSOCIATES

ACOUSTIC ENVIRONMENT SIMULATION STUDY(U)
ACOUSTIC INTRUSION SENSOR PERFORMANCE(U)

by

Loren Enochson, Robert K. Otnes

84 12 31 005

e

FINAL REPORT

ACOUSTIC ENVIRONMENT SIMULATION STUDY

PREPARED BY:

LOREN ENOCHSON
TIME SERIES ASSOCIATES
920 WEST 33RD AVENUE
SPOKANE, WA 99203

PREPARED FOR:

NAVAL SURFACE WEAPONS CENTER
WHITE OAK, MD 20910

NOVEMBER 1982

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Time Series Associates 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Naval Surface Weapons Center SNWS Project Office, Code H101
6c. ADDRESS (City, State, and ZIP Code) Palo Alto, California 94301	7b. ADDRESS (City, State, and ZIP Code) White Oak Silver Spring, Maryland 20903-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N60921-82-M6028
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 63571N · PROJECT NO. · TASK NO. · WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
Acoustic Environment Simulation Study (U); Acoustic Intrusion Sensor Performance (U)

12. PERSONAL AUTHOR(S)
Loren Enochson; Robert K. Otnes

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final; Final	FROM ______ TO ______	82,Nov; 83,Jan	24; 40

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Acoustic Control, Acoustic Simulation, and Acoustic Intrusion Sensors. Signal Processing

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

A method to perform controlled acoustic tests up to a frequency of 20-25KHZ was explored. This method is based on enhancements to the method currently used to perform random vibration tests.

Various methods of signal processing for acoustic intrusion sensors are explored and evaluated. Among the signal processing methods investigated are: Doppler Approach, Time History Approach, Power Spectral Density Approach, and the Cepstrum Approach.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Tom Peacock	22b. TELEPHONE (Include Area Code) (202) 394-1692 · 22c. OFFICE SYMBOL H101

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

TABLE OF CONTENTS

GLOSSARY OF SYMBOLS

ADS = ANALOG TO DIGITAL CONVERSION SUBSYSTEM
ASCII = AMERICAN STANDARD CHARACTER SET FOR INFORMATION INTERCHANGE
ASPEC = AUTOSPECTRUM SUBROUTINE IN TSALF
b = RESOLUTION BANDWIDTH = S/N = B/L = 1/P
B = TOTAL ANALYSIS BANDWIDTH
CSD = CROSS SPECTRAL DENSITY FUNCTION
CSPEC = CROSS-SPECTRUM SUBROUTINE IN TSALF
DAS = DIGITAL TO ANALOG CONVERSION SUBSYSTEM
EAE = EXTENDED ARITHMETIC ELEMENT
EIS = EXTENDED INSTRUCTION SET
f = FREQUENCY VARIABLE = kb
FIS = FLOATING INSTRUCTION SET
FFT = FAST FOURIER TRANSFORM
FFT-1 = INVERSE FFT
i = TIME INDEX
k = FREQUENCY INDEX
L = NUMBER OF SPECTRAL LINES = B/b
MAD = MULTIPLY-ADD OPERATION
N = NUMBER OF DATA POINTS IN TIME PERIOD
P = TIME PERIOD(SEC) = NT
PSD = POWER SPECTRAL DENSITY FUNCTION
S = SAMPLING RATE, SAMPLES PER SECOND (SPS)
SETADS = ADS SETUP SUBROUTINE IN TSALF
SETDAS = DAS SETUP ROUTINE IN TSALF
Sx(f) = POWER (AUTO) SPECTRAL DENSITY FUNCTION OF x(t)
Sxy(f) = CROSS-SPECTRAL DENSITY FUNCTION
t = TIME (SEC) = iT
T = SAMPLING INTERVAL(SEC) = 1/S
TFAM = TRANSFER FUNCTION ANALYSIS MODULE
TSALF = TIME SERIES ANALYSIS LIBRARY: FORTRAN
x(t), y(t) = TIME HISTORIES
X(f), Y(f) = FFT'S OF x(t) and y(t)
* = FORTRAN MULTIPLY

1. MARKET SURVEY

Manufacturers of existing shaker control systems were surveyed to determine the availabilty of an acoustic environment simulation system. The manufacturers surveyed wereGenRad, Vibration Analysis Division: Hewlett Packard: Zonic Corporation: MBIS, Inc.: and Scientific Atlanta. A letter was sent to each except GenRad from whom information was previously available. Replies were received except from Scientific-Atlanta.

Copies of the responses received are included as an appendix. The responses were negative in terms of being able to accomplish 20-25 KHz acoustic environment control. These respones were expected. MBIS claimed 5KHz control and 15KHz analysis capability. The other systems are good to about 2-4Khz bandwidth in shaker control. Most systems have a 25KHz bandwidth analysis capabilty.

The GenRad 2500 series systems possess an 8KHz bandwidth for closed loop shaker control and 25KHz for general data analysis. GenRad has supplied two acoustic chamber control systems in the past with bandwidths of 8KHz. The digital to analog conversion subsystem (DAS) of the GenRad system has an inherent bandwidth capability of 25Khz, the same as its ADS (analog to digital subsystem). These capabilities plus the fact that several GenRad systems are already in operation at NSWC in related applications leads to the selection of a GenRad system as being most suitable for detailed evaluation. Thus future compatibilty problems are minimized while providing a hardware system capability at least as powerful as the others. In addition, since GenRad systems are PDP-11 computer based, the door is opened to possible substitution of higher performance computers or the addtion of moderately priced array processors to increase the performance of the system.

The specific model evaluated is the highest performance GenRad Signal Analysis System available. A DAS is added to the system to provide the acoustic data generation capability necessary for acoustic environment simulation. The system model number is GR 2500-9701. The basic characteristics are:

* 4 channel ADS
* 1 channel DAS
* 64K 16 bit words memory
* CDC Hawk 10megabyte hard disk
* PDP 11/34A processor
* GR CRT terminal and hard-copy printer plotter

Timing estimates were also made with a PDP 11/44 processor substituted for the PDP 11/34A. Dual floppy disks are a desirable option also since they provide a more convenient media for transmitting software than the rigid disk cartridge.

2. SYSTEM LIMITATIONS

The existing GR system performs its computations in a PDP 11/34A and a special GR 2501 microprocessor. The closed loop shaker control system performs certain computations in its standard form which serve to limit the overall speed and thus the bandwidth capability of the system. The main tasks are:

1. Continuous DAS output.
2. Computation of DAS output buffer random starting address.
3. Overlap tapering of output data blocks.
4. Acquisition of control accelerometer data.
5. Computation of averaged PSD via FFT.
6. Computation of exponentially averaged PSD.
7. Determination of alarm and abort conditions.

The present GenRad system can input and output data at a rate of about 20,000 sps. This provides a basic time interval of

$$T = 1/20000 = 50 \text{ microsec}$$

In a typical frame size of N = 1024 (which corresponds to L = 400 usable lines of frequency resolution across the total analysis bandwidth B) one has a time period of about

$$P = 1000(.000050) = 50 \text{ millisec}$$

available for the various computational tasks that must be performed.

Within the 50 microsec interval the system must output data to the DAS and input data from the ads. In addition there must be adequate time to perform the seven major tasks listed above. These tasks can be performed in segments as time is available from each of the basic 50 microsec intervals.

The acoustic control systems delivered by GenRad in the past were implemented on older, but similar, GR (X2) systems. A maximum data rate of about 20,000 sps was attained in these systems also.

Although the DAS has the capability to operate at faster rates, the 50 microsec interval is the practical minimum in a standard GR 2500 series system. This is due to the fact that that the GR2501 microprocessor also is employed to emulate the DEC Extended Arithmetic Element. The most time consuming instruction which is emulated (the divide instruction) takes about 30microsec. During this interval the Unibus is locked out and no other device may access it. All devices including the ADS and DAS need access to the Unibus for every data word that is input or output. Thus when the EAE emulator is employed, the absolute maximum data rate is about

S = 1/.000030 = 33,000 sps

This implies that the EAE emulator must not be used for the acoustic environment simulation. This turns out to be an advantage in most respects. The EAE emulation in the GR 2501 microprocessor is one of its least efficient tasks and the alternatives described below are all more efficient with very little cost penalty.

Three alternative possibilities exist:

1. Use floating point arithmetic with a floating point arithmetic processor.
2. Use an 11/44 with faster standard fixed point arithmetic.
3. Utilize an 11/34A processor with an EIS (Extended Instruction Set) and perhaps 'cache' memory.

In the interest of minimizing the development of any new software, the use of existing GR software is definitely advantageous. One module called TFAM (transfer function analysis module) employs the DAS and outputs certain types of specified time series functions. However, standard capabilities of this package do not allow for the output of a general random noise process of the nature required for acoustic environment simulation. The basic DAS driver subroutine, or its counterpart in the Fortran TSALF package should be highly useful. The GR TSALF package is a set of subroutines that performs the most useful functions of Time Series Language. This subroutine set is very useful in generating Fortran programs which can easily access the special microcoded features of the GR 2501 microprocessor. These routines include the FFT, ADS and DAS drivers and interactive display routine.

3. RECOMMENDED CHANGES

In this section, the effort necessary to adapt a GenRad system to the acoustic environment simulation problem will be described. To better understand the signal analysis aspects of acoustic environment simulation, we shall first describe what TSA believes to be the ideal system (more or less) with minimal regard to cost. Then the typical cost effective methods that are employed in commercially available shaker control systems will be described. We shall find that the commercial approximations to the ideal system can be useful for generating a basic acoustic environment simulation system but the more advanced system will require the ideal system approach.

3.1 Ideal System

This system would generate an environment defined via a frequency domain function. This function would be transformed to the time domain via an inverse FFT. Incoming data would be analyzed in the frequency domain and an averaged PSD would be computed. This averaged spectrum would be ratioed with the previous output drive spectrum. This ratio would be multiplied by the reference (environment) spectrum which would be used to generate a new drive. In this way differences are corrected out and the response spectrum is forced to be equal to the desired environment. A basic PDP 11 system cannot accomplish this due to the heavy computational load. An array processor added on though can add adequate computational capacity.

A main feature of this approach is the use of a digital random number generator (sometimes called pseudo random number generators since they repeat after a large number of points -- typically about four billion). A random phase function is generated in the frequency domain. This is combined with the reference amplitude spectrum via a polar coordinate transformation to obtain real and imaginary parts which are then inverse transformed to the time domain. This is done for every data block in the ideal system. The computational load is heavy and the recipe is as follows:

1. Generate random phase function.
2. Combine with reference magnitude and convert to real and imaginary parts to obtain the complex frequency function X(f).
3. Pad X(f) with N zeroes and compute double length inverse transform to obtain time history segment. (This zero padding is necessary to avoid what is known as 'wraparound' error and to allow the generation of a continous true convolution).

$$x(t) = FFT^{-1} [X(f)]$$

4. Overlap and add the first half of the new x(t) segment with the second half of the previous x(t) segment.
5. Store in ADS buffer for output.

6. Input y(t) and compute Y(f) = FFT [y(t)].
7. Compute Sy(f) and average.
8. Compute Sx(f)/Sy(f) and multiply by refernce spectrum to obtain new drive spectrum magnitude. Repeat.

This method requires that an FFT be computed for every output data frame. The input data frame can be acquired and the FFT computed every data frame also but little is lost if the system is reasonably stable and data is missed. The averaged spectrum will still be representative even though data is skipped. The FFT computed for the output therefor becomes the big time consumer.

A complex FFT requires about 4NlogN multiply-adds to compute. Suppose N = 2**10 = 1024. Then the number of multiply-adds (MADS) is

No. MADS = 4X1024X10 = 40,960

This complex FFT for 1024 points would allow for zero padding. If data must be output at a rate of 51,200 sps to allow for a minimum 20KHz bandwidth, then we need

51,200/1024 = 50 frames/sec

The number of MADs/sec then is

MADs/sec = 50X40,960 = 2,048,000

Thus we need better than a two MegaHz multiply-add rate. This implies either a large computer or a PDP 11 augmented with an array processor is required. An array processor such as an Analogic AP400 has a rate of about 5 MegaHz. Additional capacity would be required for the random number generation and other tasks, but it appears possible to accomplish the task.

Basic driving software in the form of Fortran callable routines are available from Analogic which will greatly simplify the preparation of special software when this array processor is used. The price of this software package is included in the cost estimates presented later.

3.2 Practical Commercial System

Several approximations are made in the control algorithms employed in commercially available systems.

1. No zero padding to accomplish the overlap-add convolution is performed. Instead, the data frames are overlapped by 50% and added together after tapering the ends with a cosine bell.

2. A new random frequency function is not generated every data frame. This is only done once every several data frames de-

pending on bandwidth, error checks, plotting and other processing parameteres. The inverse FFT is only performed when the new data frame is generated. The data frames in between are somewhat randomized by selecting a random starting address in the DAS ouput buffer, and the data frame is treated in a 'wrap-around' manner. The overlap and taper is then performed as described in (1) above. This and other features are presumably protected by GR patents. GR would allow the use of these techniques in any special software prepared to operate on one of their systems. One would not be allowed to commercially market such a system however.

3. The frequency function does not have a unif[illegible] stributed random phase, but rather a phase with +/[illegible] and +/- 90 degrees randomly selected. This is obtained by randomly switching signs on the magnitude which amounts tc [illegible]aining sines and cosines of these angles except for a scale factor of 1/(2**1/2) which is accounted for later.

The above approximations have proven adequate in practice and allow more rapid speeds and higher bandwidths than would otherwise be possible. As discussed below we would employ this approach for a baseline system. However, the more sophisticated system would require the more complicated approach.

A significant amount of computer time is spent on various safety features such as alarm, abort and loss of signal checks. Also if plots are requested by the operator to monitor the progress of a test, then a considerable load is placed on the computer and loop times are slowed. (A loop is defined as the amount of time between generation of a new control spectrum and the inverse transform to obtain a new random time domain function).

Occasionally 'pseudo random' control is employed. The term in this context means that a random appearing time function is generated that possesses a given discrete spectrum. The problem with such functions is that they have a time period which is the data frame length and thus possess a line spectrum. Hence, no power exists in between the spectral lines and it is difficult to predict what will happen at these frequencies. The same function is generated each time and thus has 'zero variance' except for extraneous noise. However it is generally an unsatisfactory approach because of the lack of excitation power between spectral lines.

3.3 Recommended System Changes

It would appear that any control system which is to attain 20KHz bandwidth or higher must use the approximations described in Section 3.2. However, augmenting the system with an array processor such as the Analogic AP400 may be cost effective by simplifying algorithms and

some of the programming problems. That is, the speed allows for simpler and more easily implemented methods.

The GenRad system employs special microcode in certain parts of the shaker control algorithm. These are the various functions with their estimated execution speeds:

1. FFT - 75 millisec for N = 1024 points.
2. Auto and cross-spectrum averaging - 16 and 50 millisec respectively.
3. Random number generation - exact timing unknown
4. Overlap-taper - 200 points in about 8 - 13 millisec.

The FFT and spectrum averaging are accessible via either TSL or TSALF routines. The random number generation and overlap-taper and tightly integrated into a very special purpose shaker control package. Thus they are not accessible by normal coding methods. Extremely detailed knowledge of the GR2501 microprocessor and the shaker control software is necessary in order to access and modify these routines to make them suitable for acoustical control. Due to these considerations, it does not seem practical to make use of the overlap-taper and random number generation routines. It will be necessary to prepare counterparts of these routines for the recommended AP400 array processor.

The following modifications to the shaker control algorithms are recommended to increase the bandwidth:

1. Increase the frame size to the maximum possible, probably N = 8192.
2. Reduce the tapering and overlap to 1/8th or less of the length of the data frame. It is believed that this amount of overlap is adequate to avoid frame discontinuities and although the technique has not been thoroughly tested it is believed that it wouold entail any significant technical risk.
3. Eliminate alarm and abort tests except for overall RMS and loss of signal.
4. Add and EIS or the equivalent and eliminate the need for the 2501 to emulate the EAE. As mentioned previously this does not give up but gains performance at a small price penalty.

Since access to the 2501 microcode is not possible, the above recommendations are not feasible to implement by direct modification of the existing routines.

Time estimates with machine language routines indicate that a PDP 11/34 might be able to accomplish the task if augmented with cache memory and if the EIS or an EAE were used. The code employed to accomplish this estimate is as follows:

```
                                        TIMING - microsec
      SETUP   MOV     COUNT, R4         1.96+1.46+.93 = 4.32
              MOV     AD3, R3                           4.32
              MOV     ADR1, R0                          4.32
                                                TOTAL  12.96
      LOOP    MOV     ADR2, R1                          4.32
              MOV     (R0)+, E15                        4.32
              MOV     (R1)+, E15                        4.32
              MUL     E15                8.95+1.89     10.84
              ADD     E15, (R3)+         2.16+1.89      4.05
              DEC     R4                 1.96           1.96
              BLE     LOOP               2.31           2.31
                                                 TOTAL 32.12
```

Floating point times for an 11/44 are estimated to be slightly greater at about 35.5 microsec. An 11/44 in fixed point is estimated to be about 15.5 microsec. The addition of cache memory to the 11/34A may reduce the 32.2 microsec to almost half, about 17 microsec, if the routines are carefully coded with cache in mind.

If N = 8192 point frame sizes are assumed with 1/8 th overlap, then 2048 points must be tapered and added each of these frames. The time available for each frame assuming 50KHz bandwidth and S = 51,200 sps is

$$Ps = 8192/51200 = 160 \text{ millisec}$$

The total time for overlap tapering is

$$T\ taper = .017 \times 2048 = 34.82 \text{ millisec}$$

This would appear to leave time for random number generation, general overhead, spectrum averaging, control algorithm and a safety factor. However it is felt that the system would fast run out of capabiltiy if higher bandwidths were attempted.

The major recommendations are the following:

1. Do not attempt to use the existing GenRad shaker control package.
2. Use machine language (or array processor) routines for tapering and random number generation.
3. Use the FFT, ASPEC and CSPEC subroutines from the TSALF package for power(auto) and cross spectrum computations.
4. Use TSALF ADS driver SETDAS as discussed in Section 5.
5. Use 8192 point frame size with 1/8 th overlap.
6. Minimize alarm and abort testing.
7. Use closed loop for initial establishment of control, but perform open loop testing after that.

It is believed that a rate of about 50,000 sps can be attained with

these procedures. It is further believed that augmentation of the system with an Analogic AP400 array processor will simplify implementation of the overall system and allow expansion capability. In addition, higher bandwidth control will require the array processor.

Higher data rates (past 25Khz bandwidth) will encounter the rolloff of the all-pass setting of the DAS filter. It may be possible to compensate this by adjustment of the control spectrum. However detailed information on the DAS output filter is not available to determine the exact effects.

4. COST OF CHANGES

The estimated cost of the recommendations in Section 3 are as follows:

1. TSALF Package	7,000.00
2. AP400 Array Processor with necessary memory, PDP-11 interface and Fortran callable soft ware package.	16,000.00
3. Cache memory.	3,000.00
4. Coding of overlap taper and random number generation for AP400. Four man weeks at 2500/man week.	10,000.00
5. Coding of main control routines in Fortran and TSALF. Twelve man weeks at 2500.	30,000.00
6. Documentation and system integration. Two man weeks at 2500.	5,000.00
7. Travel, per-diem and miscelaneous expenses.	5,000.00
TOTAL	76,000.00

Note that these estimates include both hardware and software. Also, the TSALF package is a standard GenRad optional software item and would probably be included in a basic GenRad system.

5. USE OF FORTRAN (TSALF) ADS DRIVER

A digital to analog subsystem control routine, SETDAS, is available as a standard feature in the GenRad TSALF package. This routine is employed in the TFAM (Transfer Function Analysis Module) software. Although TFAM does not possess the necessary flexibility for direct use in the acoustic environment simulation program, the SETDAS routine is essential and directly usable.

The only proviso for use of the SETDAS routine is that the latest version be employed. Early officially released versions contained bugs that would adversely affect its use for the acoustical package. However, an internally released version does exist which is usable. This routine will probably be in official release before it is needed in the subject project.

The SETDAS routine should possess the inherent capability to output data at a rate of about 140,000 sps. This has not been tested to the best of the author's knowledge however. Thus unseen problems could exist at higher data rates, but this is considered unlikely.

It is expected that SETDAS is usable as is with no modifications at no additional cost. This routine is a standard part of the GR TSALF package which is included in the system cost estimates.

6. MULTIPLE CONTROL POINTS

The existing control loop control programs are primarily based on the acquisition of data to compute a single PSD. However, optional PSD control channels can be specified in the standard system. Up to 16 control channels are possible. This means that data from up to 16 channels will be acquired, PSDs computed, and all averaged together to form a single averaged control spectrum. This slows down the control loop since more data must be acquired and additional PSDs computed. However, for reasonably stationary systems this does not pose any special problems.

Thus, there does not seem to be any inherent problems in acquiring and averaging either two or four channels in the acoustic control system. One major difference is that higher sampling rates will be involved in the acquisition. If we assume 50,000 sps per channel, and if four channels are acquired simultaneously, then the overall data rate capability of the ADS is reached. Also, the data frame size may be limited unless additional optional memory is obtained with the system.

Some additional control accuracy is attained by averaging multiple control points but it may not be necessary. It will depend quite a bit on the geometry of the space in which the acousitcal signal is being generated. If the geometry is of sufficient complexity so that variation from point to point in the space is encountered then the multiple points will be an advantage. *Otherwise they will not be* worth the added expense.

7. ACOUSTICAL CONTROL SOFTWARE OUTLINE

This section contains an outline of the acoustical environment control software. It is patterned after the GenRad shaker control software. This is felt to be the simplest to implement. It has proven successful in shaker control and acoustic control at a lower bandwidth. It is therefor felt that there is little technical risk in the described method.

7.1 Software Outline

The elements of the acoustical control software are as follows:

1. Input test ID, date, time, operator ID.
2. Input setup file name if previously specified.
3. Edit setup file by line number if available.
4. Enter file setup dialog if new setup.
5. Enter setup information as per GenRad shaker control system manual 1923-5146. This will be accomplished on a conversational interactive basis. However the GR control panel will not be employed in any way.

   Modifications as follows will be accomplished.
   a) The input spectrum will be allowed as a narrow band spectrum with up to 32 breakpoints. Initial and final slopes will be automatically selected by the program unless overrode by the operator.

   b) The spectrum will also be allowed to be input as a 1/3 octave band spectrum in 30 bands from 20 to 20KHz. ANSI frequency bands will be employed. In either spectrum the levels can be specified in dB or KiloPascal.
6. Special features including extremal averaging, level scheduling, alarm and abort limits, and special auxilliary channels will not be included.
7. Save setup in disk file under default name or one selected by user.
8. Perform loop check. Output signal at 1/10 of RMS of specification spectrum. Compare RMS of each input channel with noise background. Alert operator if input levels do not exceed noise.
9. Initate control at low level. If control is established notify operator and proceed to high level in 5 increments on operator command.

   Control algorithm will duplicate that described on pp 3-7,8 in GenRad amnual 2501-0202 'Training Note: Random Vibration control Program'.
10. When full level control is attained, proceed to open loop data generation and input data spectrum computation. Terminate based on elapsed time or operator command.
11. The spectrum averaging process will be essentially the same as

in the GenRad shaker control program with some exceptions.
a) The data will be acquired in 8192 frame lengths or as long as possible. The time history will be tapered with a 1/10 cosine taper window. The FFT will be computed and smoothed back to specified resolution, usually 400 lines. The control process will be lengthened to allow for reverberation settling and the spectrum will possess more desirable leakage characteristics. The same control strategy will employed from this point.
b) The output data will be done in 8192 point buffers with tapering and overlap done on 1/8 th of the length rather than half as is the case in the shaker control system

12. Multiple channels of data will be acquired and used in establishing control. Also these spectra can be save on disk in the same manner as allowed for in the GenRad shaker control system. When this is done, the loop speed will be decreased.
13. Abort and alarm checks will be performed only on the overall RMS levels rather than on individual spectral lines. It is believed that these checks afford adequate safety for acoustic control applications. The problems encountered in testing and possibly damaging a mechanical structure do not exist here.

These are the major features of the software system which will follow the well tested exixting algorithm to a great extent. Changes have only been made to accomodate the higher bandwidth data and the slight differences between vibration and acoustical data.

The performance of this basic software should attain 20KHz and possibly 25KHz bandwidth. It is nearly impossible to define the exact limit without running some simple simulations to determine the effect of the cache memory and the capability added by the AP400 array processor. The software in its initial configuration is intended to perform closed loop functions only in the initial establishment of control. Then the subsequent processing would be performed on an open loop basis with no further adaptation to any changes in the environment. For a single room under stationary conditions this should be satisfactory. It is likely that the AP400 possesses adequate capacity to later add this more complex feature however. It is also likely that the AP400 will provide the capacity to later expand the bandwidth to 50KHz. However there remains the unknown of the performance of the DAS output filter in its all-pass mode which is necessary to determine. This will affect the shape of the environment spectrum in a presently unknown way.

The cost of preparing this initial baseline version of the acoustic environmenmt control software is the same as described in Section 4.

8. FUTURE ENHANCEMENTS

Two future enhancements are considered in this section. The first is the introduction of acoustic transients into the acoustic environment. The second is the performance of statistical record keeping to determine the performance of acoustic intrusion sensors.

8.1 Acoustic Transient Generation

The object of this enhancement is to allow the system to generate acoustic signals that would simulate intruders entering a room and introducing changes to the acoustic data received by the system ADS. One implication here is that there should be no closed loop control when this is done. Otherwise, the system would attempt to modify the output data it was generating and change the character of the background environment to compensate for the transient.

There are two basic ways to generate these transients:

1. Generate a time domain signal and add to the time history before the DAS output, and

2. Modify the Fourier transform that is used to generate the frequency domain version of the acoustic environment before it is inverse transformed to the time domain.

The first way seems the most straightforward. Assuming that it is possible to determine the acoustic waveform of footsteps or the like, then this waveform could be generated and stored in memory. The waveform could be recalled at appropriate time intervals and added to the output data buffer prior to output by the DAS. The disadvantage to this approach is that it is a completely new and additional computational task that must be added to the computer system. However the array processor may provide adequate capacity.

The second way is in principle about as straightforward. If we can add a signal to output waveform then this sum of signals can be added in the frequency domain also. The Fourier transform of a sum is the sum of the transforms. That is

$$FFT\ [\ x1(t) + x2(t)\ ] = X1(f) + X2(f)$$

In this equation we could view $x1(t)$ as the random acoustic environment and $x2(t)$ as the desired transient. Thus we can determine the inverse transform of the desired time domain waveform. Then we can add this to the randomized spectrum prior to the computation of the inverse transform to determine the output random time history. We note that this must be a complex addition in the frequency domain. That is, both the amplitude and phase of the spectrum must appropriately modified.

A drastic implication to either of these methods is that following:

* it is not possible to utilize the simpler commercial version of the random vibration control software.

The randomized rotation and overlap tapering of the output data segments would completely destroy the character of the transient waveform. This would occur whether or not the data was added to the time domain signal or the frequency domain spectrum. Thus the capacity of the array processor is absolutely essential before this feature could be incorporated into the system.

8.2 Statistical Record Keeping

When acoustic intrusion sensors are being tested by exposing them to the acoustic environment being generated by the system, it would be desirable to have the computer monitor their performance. This would seem to be possible if the the sensors can have a detect alarm signal output which is routed to the computer. It is then necessary to have additional input devices for the computer so that these signals can be monitored.

There are at least three possible devices to handle these input signals. The most flexible and probably the most expensive is to expand the ADS to a larger number of channels. The maximum available is 16. This would be overkill for the sensor detect signal, but would provide the most power which might be useful in other applications. GenRad has also provided on a semi-custom basis a lower cost 32 channel MUX referred to as the ADAC unit which is the second possibility It can be employed in a manner similar to the extra channels on the ADS but it does not possess nearly as many features. The third possibility is the alarm and abort line feature provided as an option in the GenRad shaker control system. In this hardware an interface and 16 lines are provided which can be tested for on or off. This is probably the most reasonable approach and the least expensive. The sensors could undoubtedly be constructed to supply the appropriate signal. The only statistics that could be kept in this case would the number af detect alarms generated. The more expensive multichannel inputs could probably also accept amplitude levels and keep more sophisticated statistics. In any case, it is not anticipated that this would add a severe load to the array processor augmented system.

GenRad treats the ADAC subsystem as a semicustom option and will provide these in special systems.

10. LINK WITH VAX 11-750 COMPUTER SYSTEM

A recently released GenRad product called DATALINK provides a reasonably simple method of interfacing with VAX and similar computers. This is a hardware/software package consisting of a modem and interface plus a software package. The software provides reasonably simple interface between RT-11 and RSX-11 DEC systems. Files of data in both ASCII and binary formats can be transmitted between the different DEC machines. Thus the mechanics of transmitting data files between a VAX and GenRad machine are now straightforward.

Possible uses for the link would seem to be restricted to downloading of test setups from the VAX to the PDP-11. Also final results and statistics kept during the test could be transmitted back to the VAX for detailed analysis. It is not envisioned that the VAX would play an on-line role during the running of an actual test however. It is felt that this would just constitute an additional complication that would lower the overall bandwidth of the system and probably result in no real additional benefit.

11. SYSTEM RECOMMENDATION

The system recommended has been summarized in Section 1. It is reviewed in more detail here.

ITEM	ESTIMATED COST
1. GenRad 2502 Signal Analysis System expanded to a minimumof 64K words memory and preferably 128K if available. This includes the 10Megabyte CDC Hawk disk, the PDP 11/34A processor and the GenRad CRT and printer/plotter terminal.	100,000.
2. Digital to Analog Subsystem (DAS).	8,000.
3. Shaker control alarm and abort lines as an option for statistical record keeping of acoustic sensor responses.	3,000.
3a. ADAC A/D subsystem as an option to perform statistical record keeping of acoustic sensor responses.	12,000. (estimated)
4. Dual floppy disk option.	8,000.
5. DATALINK option. (Option for VAX communication.)	9,000.
6. TSALF and Fortran software packages.	7,000.
7. Cache memory. (non GenRad option.)	3,000.
8. Analogic AP400 array processor (non GenRad option.)	16,000.
9. System integration.	5,000.
10. Software for basic system.	50,000.
11. Software for revised system to perfrom more sophisticated time history generation involving the methods described in Section 3.1.	25,000.
12. Software for generation of transients.	10,000.
13. Software for stastistical record keeping.	7,500.

Note that no allowance is made here for additional racks or backplanes that might be necessary to accomodate the various hardware options.

It is felt that it is necessary to initially procure the most sophisticated hardware except for the modem and interface in the DATALINK package and the alarm and abort line for statistical record keeping. The fastest system is needed for even the 20-25Khz bandwidth analysis. Only the software will be simplified and based on more thoroughly tested procedures in the initial system. Thus there is very little technical risk in the baseline system and software.

The integration of the AP400 into the GenRad system involves some technical risk in that it has not previously been accomplished. However the AP400 is specifically designed with DEC computers in mind and it has been extensively used in system involving DEC computers.

The cache memory is not a standard option, but it is believed to be a quite simple addition to a system involving nothing more than inserting a card into the chassis. The only potential problems are that additional rack space and additional backplane space might become necessary as the various option are included.

There does exist a problem involving warranty and service of the equipment. GenRad will undoubtedly not provide the entire integrated system involving the nonstandard GenRad equipment. Thus NSWC must be prepared to perform this integration and service in an alternative manner. There does not seem to exist any particular problem with - complising this since existing personnel at NSWC have the necessary experience and qualifications.

The expansion past the 20-25KHz bandwidth requires more careful attention to optimizing the coding and probably tranferring additional computational load to the array processor such as the FFT and the PSD averaging. The interface between the AP400 and the GenRad system becomes slightly more complex at this point and the technical risk is increased. An additional unknown at this bandwidth is the performance of the DAS filters is the all-pass mode to allow 50KHz data generation.

A final option that could be useful for software development purposes would be a DEC VT100 terminal and DL11 interface card. This would allow the use of more sophisticated full screen editing software which is very useful in preparation of Fortran programs.

11. SUMMARY AND CONCLUSIONS

This study has resulted in the evaluation of the applicability of a GenRad signal analysis system to acoustic environment simulation. Gen Rad hardware and software has been evaluated as being the closest to providing a system that has the necessary performance. In addition the fact that other GenRad systems exist at NSWC leads to operationsl simplication in that new software and procedures need not be learned by cognizant personnel. Various key points of this study are summarized below.

1. The GenRad signal analysis system provides the optimum base.

2. It is not possible to use existing GenRad software packages for acoutical control above about 8,000 Hz bandwidth. It might possibly operate at 10-12KHz but special testing would be required to determine the upper limit.

3. It is not feasible to modify existing high performance microcoded routines which are used in the shaker control package so that the bandwidth can be increased. This would require a separate computer for firmware development plus the capabiltiy of burning ROMs. In addition, detailed knowledge of the highly specialized GenRad microprocessor is required. The shaker control software is a tightly coded highly optimized package to accomplish a very special purpose and any modifications to it involve tremendous technical risk.

4. The considerations in item 3 lead to the recommendation of the use of the Analogic AP400 array processor as a fundamental part of the system.

5. The basic algorithm employed in the GR shaker control package is recommended as the baseline software. In this package the AP400 would perform slightly modified overlap-taper routines where a longer frame size is employed to speed the processing. It is recommended that a new special Fortran program be prepared.

6. The addition of cache memory to the system appears to be a low-cost, low technical risk option that can provide considerable additional speed if planned for in the program.

7. A recently released package called DATALINK from GR can make communication with the VAX 11/750 a relatively simple task. However, there does not appear to be a large role that would be played by the VAX in this application.

8. Multiple control points can be fairly easily handled. This might improve performance with little degradation in speed.

9. The implementation of a capability to the system that would in-

volve generation of acoustical transients would require a significant software upgrade to the system. A much more sophisticated approach is necessary in this case. The AP400 would play a much bigger role in this case.

Some uncertainties exist which it would be desirable to test and evaluate. This includes the effect of adding cache memory and the performance of the ADS filters above 25Khz. Time Series Associates posseses a GR 2510 which has the standard ADS filters and these could be tested. This would require either an accurate signal generator good out to 50Khz or a white noise generator which is guaranteed flat out to this frequency. The 2510 would not serve for the memory test but a 2500 series system with a PDP-11 could easily be rented at a moderate cost to perform timing tests. These tests would require about one man-week at a cost of $2500.00 plus the use of a computer for two days at a cost of $1000.00.

The implementaion of an acoustic environment simulation system seems to be a very feasible task at the bandwidths desired. Although the. hardware available seems to possess adequate inherent cpacity, the existing software does not and special programs will have to be prepared. Overall the technical risk seems moderate, and it seems to be an logical approach to the accomplishment of such a task.

ACOUSTIC INTRUSION SENSOR PERFORMANCE

CONTRACT N60921-82-M6028

FOR THE

NAVAL SURFACE WEAPONS CENTER

ROBERT K. OTNES, PHD

TIME SERIES ASSOC.

2160 Middlefield Road, Palo Alto, California 94301, (415) 324-1821

0. TABLE OF CONTENTS.

0.1. TERMINOLOGY

ATSA	APPLIED TIME SERIES ANALYSIS, R. K. Otnes and L. Enochson, Wiley Interscience, 1978.
b	Frequency interval of the DFT in Hz.
CW	Continuous Wave. A sine wave always on.
df	Incremental Doppler shift in Hz.
f	Frequency in Hz.
F	The Nyquist folding frequency = S/2.
FAR	False alarm rate = M*P10.
Hz	Hertz, the unit of frequency.
H0	No intruder hypothesis.
H1	Hypothesis that there is an intruder.
H(f)	Effective transfer function of the room.
M	Number of tests per unit time.
MR	Miss rate = M*P01.
N	Number of points in record being processed.
p(i)	The time history of the generated signal.
P	Record length in seconds = N*T.
PSD	Power Spectral Density. See S(k).
P00	Probability H0 chosen and H0 occurred.
P01	Probability H0 chosen and H1 occurred (probability of a miss).
P10	Probability H1 chosen and H0 occurred (probability of a false alarm).
P11	Probability H1 chosen and H1 occurred.
q	Index of reflections.
Q	Total number of reflections.
r	Detection test statistic: result of processing.
R	Hypothesis threshold. If exceded, it is assumed that there is an intruder.
s	Speed of sound in air in feet per second (1087 fps).
sps	Samples per second, the rate of sampling data.
S	Sampling rate in samples per second, generally assumed to be 80 KHz in this application.
S(k)	Power spectral density at k*b Hz.
t	Time in seconds.
T	Sampling interval in seconds.
v	Velocity of an intruder relative to the sensing devices.
x(i)	Time history of data from microphone.
X(f)	Fourier transform of x(i), continuous frequency.
X(k)	Fourier transform of x(i), discrete frequency.
y(i)	The Hilbert transform of x(i).
z(i)	The envelope of x(i).

1. INTRODUCTION.

The purpose of this report is to investigate what acoustic signal processing is required to optimize the performance of acoustic intrusion sensors for a specific application on board ship.

In particular,the report considers the following aspects:

1. Evaluation of the type of pulse that an active ultrasonic sensor should emit. Considered are: a. A single tone 20 KHz, 1 ms pulse. b. A narrow band swept sine. c. Three tones of different frequencies. d. A narrow band random burst.

2. Review of the existing acoustic spectrum from typical shipboard spaces so that typical spectrum levels in the detection band can be determined. Not much of this type of data was available.

3. Investigation of special signal processing techniques to achieve a high probability of detection and a low false alarm rate. Determine if processing should be done in the time or frequency domain. Determine if computing the power spectral density function of the received waveform and comparing it to a stored power spectral density would be a useful technique. Investigation of using signal processing techniques such as the Hilbert transform and the cepstrum.

Section 2 reviews the classic detection problem for the purpose of introducting the standard terminology. Section 3 discusses the nature of the acoustic signals from several points of view. Section 4 makes reference to Doppler processing (not being implemented in this system: the information is included for completeness). Section 5 discusses processing algorithms strictly from an implementation point of view. Section 6 takes up the statistical problems. Section 7 has an example of FORTRAN code for the time averaging algorithm selected. Finally, Section 8 has the conclusions, reccomendations and a suggested course of action. The conclusion is that the type of system considered looks very feasible. The recomendations include a plan for collecting data for verification of the procedures.

The terminology used in this report is essentially the same as that in APPLIED TIME SERIES ANALYSIS, VOLUME I, by Robert K. Otnes and Loren Enochson, Wiley Interscience, New York, 1978. This text is referred to as ATSA hereafter.

2. DEFINITION OF DETECTION TERMS.

The purpose of this section is to review the basic detection problem and define the terms that are used in this report.

The detection problem consists of choosing between two hypotheses:

H0 No intruder present.

H1 Intruder(s) present.

The choice is made by examining processed data. Let us assume that the output of the sensor is run through a series of algorithms and that from these algorithms a test statistic r is produced. The number r could be generated in a number of ways. Section 5 discusses the candidate methods and algorithms. For the general discussion of this section, it is sufficient that data was acquired and processed and from that r was output from the calculations.

Associated with the test parameter r is a predetermined threshold value R. The computed r is compared with R and a decision made:

If the absolute value of r is greater than R, then choose H1, intruder present.

If the absolute value of r is less than or equal to R, then choose H0, intruder not present.

That is, the expected value of the parameter r is going to be zero. If there is an intruder, the tendency should be for the absolute value of r to become quite large, generally much larger than R.

Each time this test procedure is done, there are four possible outcomes:

1. H0 is true and H0 is chosen. No Intruder.

2. H0 is true and H1 is chosen. False Alarm.

3. H1 is true and H1 is chosen. Detection.

4. H1 is true and H0 is chosen. Miss.

The setup of this or any other detection system is done on the basis of probability. Define the probabilities of the above events as follows:

P00 = Probability that H0 is chosen given H0 occurred.

P10 = Probability that H1 is chosen given H0 occurred.

P11 = Probability that H1 is chosen given H1 occurred.

P01 = Probability that H0 is chosen given H1 occurred.

Thus, P10 is the probability of a false alarm, and P01 is the probability of a miss. There are some simple relations between these terms:

P00 + P10 = 1

P11 + P01 = 1

Suppose that during a given period of time that M such tests are made. Then the miss and false alarm rates are given by

FAR = False Alarm Rate = M*P10

MR = Miss Rate = M*P01

For example, if a test is made once per second, then during the period of one month,

M = 60*60*24*30 = 2,592,000

tests are performed. If a FAR of three (3) or less per month is desired, then P10 must satisfy:

P10 less than 3/2,592,000 = 1.1574E-6

That is, the probability of an individual false alarm in this example must be of the order of one in a million for the false alarm rate to be held to three per month when the test rate is once per second.

A standard example of this type of analysis is the following: suppose that the function x(i) is collected, where x(i) could have one of two possible forms:

x(i) = n(i) Under hypothesis H0

= n(i) + a Under hypothesis H1

where i = 0,...,N-1, n(i) is a white Guassian uncorrelated random variable with zero mean and standard deviation s, and the parameter a is a positive constant. That is, under H0, there is no constant, while under H1 the constant is present, while in both cases there is noise. If there were no noise, there would not be any problem: one could simply look at a single value of x(i) and decide on H0 or H1. The presence of

the noise turns the situation into a statistical detection problem.

The maximum likelihood detection scheme consists of forming the the test parameter r as follows:

$$r = \sum_{i=0}^{N-1} x(i) \,/\, N$$

The r parameter is just the sample mean in this case. Based on the above assumptions, it can be shown itself to be a Gaussian random variable with parameters as follows:

$$\text{Variance of } r = s^2/N$$

$$\text{Mean of } r = \begin{cases} 0 & \text{if H0 is true} \\ a & \text{if H1 is true} \end{cases}$$

That is, the variance of r is the variance of x(i) divided by N, the number of samples used in computing r. The mean of r could be one of two values (0 or a) depending on which hypothesis is true. If the value of a is known (but not whether it is present or not) and if the occurrences of 0 and a are equally likely, then we would choose as follows:

Pick H0 if $r < a/2$

Pick H1 if $r > a/2$

This is what we would always do regardless of the size of s or N.

On the other hand, while s and N do not affect the manner in which the test is set up, they very much affect the results. The odds are improved if s is made smaller relative to a, or if N can be increased. In many real examples s, the standard deviation of the original data, is fixed, so that the only way to improve the odds is to increase the size of N. The probabilities of miss or false alarm can be made arbitrarily small by making N large enough.

This is a very simple example, but it has most of the elements to be found in more complicated situations:

* The test parameter r is obtained from averaging the the data (the time history x(i) in this case). Averaging, in one form or another, is at the heart of most of these schemes. The Fourier transform and the power spectrum can both be interpreted as

examples of averaging.

* The test parameter itself is a random variable. Many times it is Gaussian no matter what for the original data was like. If the expected mean and variance of the test parameter can be estimated or computed, it is possible to determine the other characteristics of the system, such as its miss and false alarm rates.

3. THE ACOUSTIC SENSOR AND THE DATA ENVIRONMENT.

The assumptions made about the system are as follows:

1. The system consists of a SIGNAL GENERATOR, a SIGNAL RECEIVER and a SIGNAL PROCESSOR.

2. The signal generator generates one of several possible signal. The leading candidate is a burst of 20 KHz, each burst lasting approximately 1 millisecond. The bursts may be as much as 1 second apart. The signal generator is a form of loudspeaker.

3. The signal receiver is a microphone which is located close to the loudspeaker (approximately one foot away).

4. The signal processor will be some form of digital processing, perhaps a micro-processer based computer or perhaps a hardwired digital device. It inputs the analog output from the microphone, digitizes it and then does the required processing, the result of which is the test parameter r. This value is compared with a prestored value R, and if it excedes it, an alarm signal is sent.

5. The speed of sound is assumed to be 1,087 feet per second.

6. The room size is about 20 by 20 feet by 10 feet high or smaller.

Next, some signal processing assumptions are made. It is assumed that the data is sampled at a rate S samples per second. There are various possibilities, but a rate of 80,000 sps per second seems reasonable. This would put a 20,000 Hz tone exactly in between zero and F, the Nyquist folding frequency (= S/2 or 40 KHz in this case). This rate is assumed throughout this report.

The digitized time history sampled at rate S (interval T = 1/S) is denoted as x(i), where the independent variable i is the time index corresponding to i*T seconds.

The Fourier transform of x(i) is here denoted at X(k), where k is the frequency index, corresponding to k*b = k/N*T Hz. The expression relating to the two functions is:

$$X(k) = T\sum_{i=0}^{N-1} x(i)\,\exp(-j2\pi ik/N)$$

The Fourier transform will be used in this effort in a variety of ways: as an intermediate step in calculating the power spectral density, as an investigative tool by itself and possibly as a part of a detection algorithm: the Fourier transform is the optimal algorithm for detecting a sinusoid. See ATSA for details and other applications.

The inverse transform is given by:

$$x(i) = b\sum_{k=0}^{N-1} X(k)\ \exp(j2\pi ik/N)$$

The inverse Fourier transform formula reverses the Fourier transform calculation. There may be slight differences, depending on the particular computer employed, due to computer roundoff but generally these are slight.

Because these formulas are implemented using the fast Fourier transform algorithm, N, the number of points in the record is usually a power of two: 256, 512, 1024, 2048, etc.

The two sided power spectral density, S(k), is defined by:

$$S(k) = \frac{|X(k)|^2}{P}$$

The power spectral density (PSD) effectively displays the variance as a function of frequency. That is, the sum (corresponding to the integral in the continuous case) of the PSD times the frequency increment b=1/P is the total power or variance. The PSD is perhaps the most important tool used in signal analysis. The reader unfamiliar with the topic is referred to Chapter 8 in ATSA.

SECTION 3.1. THE 20 KHZ TONE BURST.

Figure 3.1 shows the simulated power spectral density of a 1 millisecond pulse of the 20 KHz sine wave. A record length of 6.4 milliseconds was employed (512 data points), thus padding the time history about six to one. Note that most of the power is in a bandwidth of less than two KHz. This is a major advantage of this signal over the other ones discussed in this section: they are all broader than this pulse if the same sort of ground rules are used.

Figure 3.2 shows a similar plot. The difference is that a second, delayed pulse with lower amplitude was added to the data set, thus creating a slightly different PSD. The second pulse occurred about 3.4 milliseconds after the start of the first pulse and has an amplitude one fourth less. Note that the heighth of the first two side lobes is somewhat higher. On the other hand, it would be difficult to say that the bandwidth had changed significantly. As will be seen, the PSD computations may not be too sensitive to variation in echoes. Such being the case, PSD processing may not be the best way to detect intruders.

A useful way of modeling what is happening is as follows: suppose that there are Q reflections of the original signal p(i), and that each such reflection has an amplitude a(q) and delay d(q), with q taking on values 1,...,Q. Thus, the received signal, not including the original pulse, has the form:

$$x(i) = \sum_{q=1}^{Q} b(q)*p(i - d(q))$$

This could be interpretated as a digital filtering type of operation. Taking the Fourier transform of this equation yields the following relationship:

$$X(f) = H(f)*P(f)$$

Where

X(f) = The Fourier transform of the received signal.

P(f) = The Fourier transform of the 20 ms pulse.

H(f) = The transfer function of the operation.

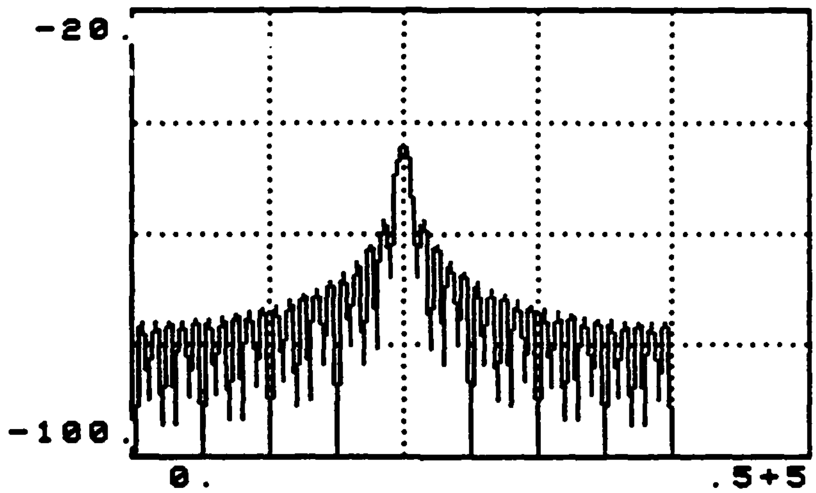


Figure 3.1. 20 KHz pulse.

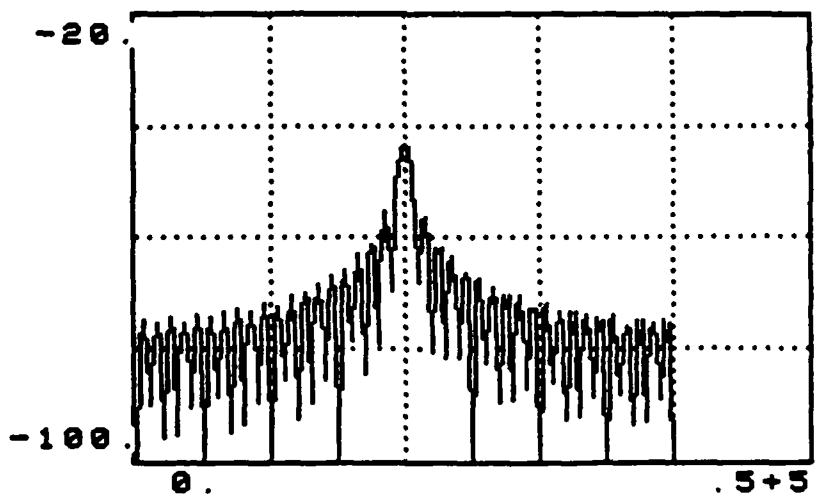


Figure 3.2. Same plus echo.

$$= \sum_{q=1}^{Q} b(q)*\exp(-j2\pi f\ d(q))$$

This model is not quite accurate, as it does not take phasing into account. On the other hand, it is adequate for the purpose of giving insight into the physical process.

The power spectral density of x(i) is the absolute value squared of this transfer function times the power spectral density of p(i). The power spectral density of p(i) is shown in Figure 3.1. Thus, what we should expect to see for the PSD of x(i) is Figure 3.1 multiplied by some sort of random transfer function (and, perhaps by the transfer function of the data acquisition system, ie., the antialiasing filter, etc.).

The transfer function H(f) is random in the sense that it is unknown until it is measured in a specific room. In theory, at least, it could take on a large number of shapes. One could imagine pathological rooms that conspire to generate a Parks-McClellan type notch filter at 20 KHz and so eliminate the signal. And in fact, such is not impossible. But it is unlikely. What is more reasonable to expect is a fairly flat transfer function with possibly some real axis zeros. Thus, while the details could and will be different than the form shown in Figure 3.1, the general outline should be nearly the same, which is to say that the expected PSD of a given room will tend to look like Figure 3.1.

In summary, the transfer functions for different compartments very likely will be very different. On the other hand, a PSD analysis of the return of a short tone burst (ie., the 20 kHz signal) will mainly see the power concentrated in a narrow band, thus making it difficult to estimate any significant portion of the transfer function, and so not giving any discrimination.

This could be overcome by making the signal broader in some manner (see the following section). The tradeoff involved, however, is probably not worth it: making the signal broader would perhaps allow a PSD/transfer function algorithm to be employed, but the broader bandwidth signals will also force the use of processing which has broader bandwidth, which in turn will tend to pick up more noise from the background. This noise will not tend to improve the processing.

Based on these observations, the algorithm choosen in this report has the following characteristics:

* It uses a narrowband signal (tone burst), so that the area where the signal is present is limited, and hence the noise that has to be processed is

limited.

* The processing is done in the time domain: Narrow functions in the frequency domain correspond to large functions in the time domain, thus making the signal processing easier.

SECTION 3.2. WIDE BAND SIGNALS.

Three types of wide band signals will be discussed:

* Swept sine waves.
* Multiple tone bursts.
* Narrow band random bursts.

Figure 3.3 shows the power spectral density in linear units of a swept sign wave. The formula for the swept sine wave is as follows:

$$p(i) = \sin(2\pi(at**2 + bt))$$

The a times t squared plus b times t term is the frequency distance of the angle: the instantaneous frequency of the sine wave is the derivative of this value, namely 2at + b. If it is required to start at time zero with frequency f0 and procede lineraly to frequnency f1 at time t1, then we must have

$$b = f0, \quad a = (f1 - f0)/2t1$$

In Figure 3.3, f0 = 17 KHz, f1 = 23 KHz and t1 = 3 ms. Figure 3.4 is the same as Figure 3.3, except that it is done in dB. As can be seen from the figures, the frequency spread is larger than the nominal six KHz spread that would be expected in this case. Also, there is overshoot of the signal at each end. It can be shown that there is always a 43 percent overshoot from the mean value, but that the relative roughness of the central part of the PSD can be made arbitrarily smooth by increasing the duration of the sweep.

If the total power in a sweep such as these is kept the same as the total power used in a tone burst, then the sweep generally will give worse results because of the broader bandwidth and the necessity of processing more of the noise.

For radar or sonar, it would be possible to use a sort of sliding bandpass filter which travels with the sweep, thus eliminating power at frequencies other than the one in the center of the filter. This type of processing would keep the noise power down.

Such processing is not possible here, as many echoes are returning with different delays. Hence, at any one time, many different frequencies would be coming in at the same time.

Multiple tone bursts have the same sort of problems that the swept sine signal has: the broader bandwidth will tend to raise the noise level in the processing, thus leading to poorer performance. The tone burst type signal does not seem to have any advantages over

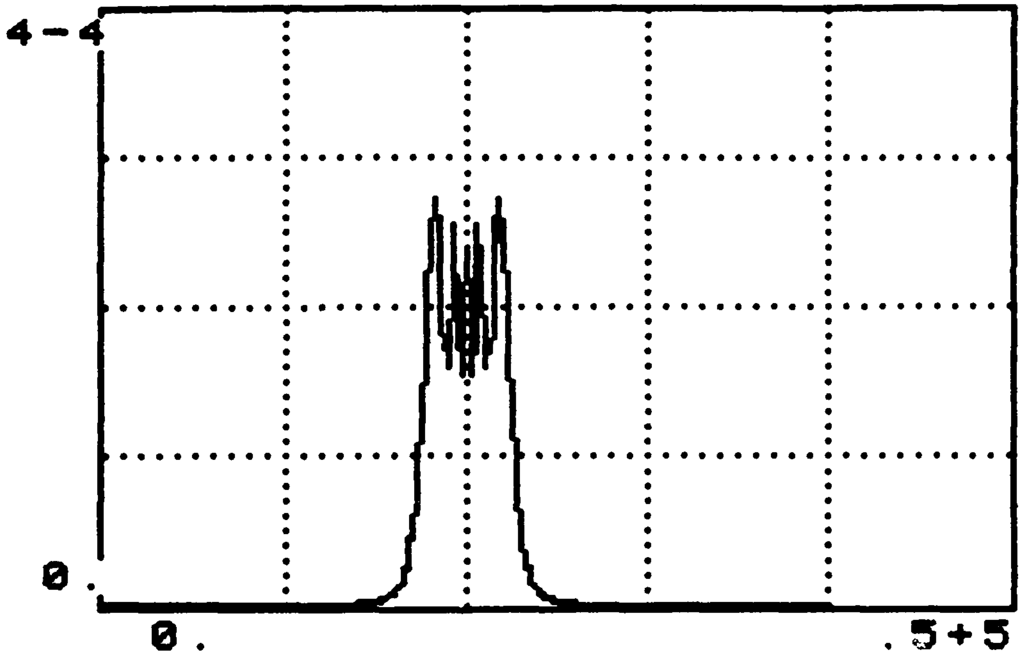


Figure 3.3. Swept sine wave.

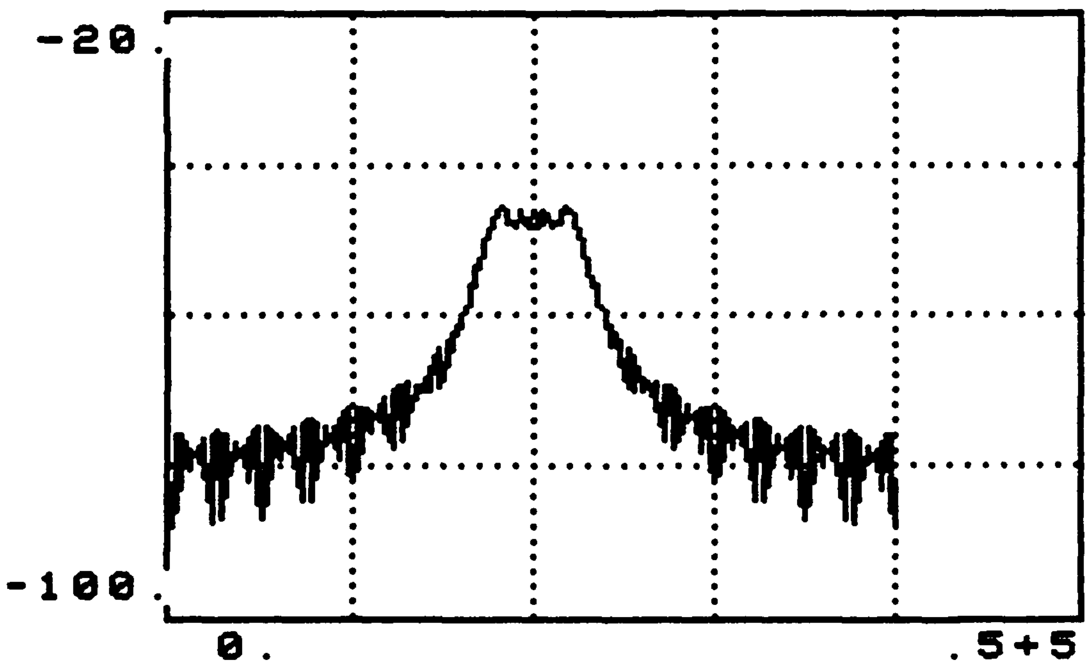


Figure 3.4. Same but dB units.

the single tone burst and the noted disadvantage.

Narrow band random bursts are another possibility: an FM (frequency modulation) scheme using ordinary noise or psuedo random noise as the modulation could be developed. Signal generators of this type are commonly used in applications where it is des ed to analyze a narrow band of frequency in the transfer function of a device. They have the advantage of having a constant minimum and maximum output as the signal they are generating is a sine wave of fixed amplitude whose frequency is being rapidly changed over a narrow range.

The random noise form has no inherent advantages over the swept sine wave in this application, while having all of the same problems.

The psuedo random noise version is of interest in super secure applications. It would be possible to have a SPEAD SPECTRUM type of device if psuedo random noise is employed. The presence and nature of such a signal is very difficult to detect and analyze even with sophisticated tools. Radar systems using spread spectrum techniques have been proposed to detect intruders in the MX missile sites. Because of their low power (50 milliwatt) and spread nature, intruders probably could not circumvent them even with the full power of a large digital signal processing facility available to them.

It requires a lot of processing to make use of such a signal. One would guess at a DEC VAX computer and an FPS AP120B array processor as being the minimum suite of equipment needed. As super security is not required here, it does not seem necessary to pursue this possibility.

SECTION 3.3. COMMENTS ON THE DIFFERENT SIGNALS.

First of all, except for pathological rooms, the tendancy of the reflections in the rooms is not to broaden out the power spectral density very much regardless of the type of signal employed.

Also, there will be background noise from many sources. One way of reducing such noise is by prefiltering the data with a bandpass filter. The narrower that the filter can be made, the more background noise that can be eliminated. For a given duration of pulse signal, the single, constant tone will have the narrowest bandwidth. So, for a given duration of pulse, the single tone appears to be the best choice.

The duration of the pulse, whatever the type of signal, also affects the results: a longer pulse can have narrower bandwidth and greater energy. For example, consider the case of a one millisecond single tone pulse and an n millisecond pulse (n an integer in this argument) of the same type. There may an n-squared gain: the bandwidth of the n second pulse will be one nth as wide, and it is producing n times the energy. With proper filtering, it might be possible to obtain substantial processing gain from increasing the pulse length, as the background noise is decreased by the filtering, and there is increased energy in the signal.

There is an upper limit on the size of the pulse: when the pulse length is so long that all of the return pulses must overlap, then probably nothing is being gained. The length relationship is also tied to the room size: the larger the room, probably the longer the pulse that could and should be used. Other parameters of the room affect the reverberation time also: the aspect of the room, the objects in it and the types of surface materials to be found there. The reverberation time of a room is simply the time in seconds that it takes for the sound level to drop 60 dB below the level it was at before all active sound producing devices are stopped. Many concert halls and theaters are into the 0.5 to 2 second range. The Illinois monument at the Vicksburg battlefield park appears to have a reverberation time of well over 15 seconds (the monument is a marble structure with a dome). A large steel surfaced room would be expected to have a long reverberation time.

In summary, the single tone pulse of duration keyed to the nature of the room seems to be the best signal to use. It will be necessary to measure the characteristics of typical magazines before an appropriate pulse duration can be established. There are a number of variables involved, the absence of knowledge of which makes it nearly impossible to estimate an optimal value for the pulse length. There are many other types of data that will have to be gathered in addition to making an estimate of the proper length of the pulse, so that such a measurement can be fit into

a test plan without any increase in its scope.

Other data that should be measured includes:

- PSD's of typical magazine background noise.
- Response of different magazines to different signals, including reverberation time.
- False alarm statistics.

4. DOPPLER CONSIDERATIONS.

It is not anticipated that a Doppler algorithm will be used in this application. Doppler based algorithms are used in many commercial alarm systems, and might at first seem to be a natural choice for this one.

A continuous wave (CW) signal is employed in Doppler systems rather than the pulse type signal discussed in this report. This section is included for completeness and to provide the reader with background material on the subject of Doppler intruder detection.

The basic Doppler equations is:

$$f' = f*(s+v)/s$$

where

- f = The original frequency before Doppler shift.
- f' = The frequency after Doppler shifting.
- s = The speed of sound in air, here taken to be 1087 fps.
- v = The velocity of the intruder in the direction of the mircrophone in feet per second.

The incremental shift, df, is given by:

$$df = f'-f = f*v/s$$

Some typical speeds and their corresponding incremental Doppler shifts are:

TYPE	SPEED	DOPPLER SHIFT
Auto at 60 mph	88 fps	1,619.14 Hz
Four minute mile	22 fps	404.78 Hz
Jogging	8.8 fps	161.91 Hz
Normal walking	4.4 fps	80.96 Hz
Slow walk	0.367 fps	6.75 Hz
Very slow walk	0.05435 fps	1. Hz

These all assume s = speed of sound in air = 1,087 fps and a value for the frequency of 20,000 Hz. The also assume that the intruder

is either walking directly towards or away from the system.

At the slow walk rate it would take nearly 55 seconds to cross a 20 foot room: at the very slow walk rate, it would take over six minutes.

If the slow walk and very slow walk rates are to be searched for, then it would be necessary to have as much as 0.1 to 1.0 seconds worth of data to process. This is on the order of 25 to 50 times more data than would be processed using pulsed algorithms, but probably still capable of being implemented with a system built around a microprocessor and a single TRW multiplier.

A typical type of algorithm for processing a CW Doppler signal would consist of down shifting the data in the neighborhood of 20 KHz to 0 Hz using a ZOOM technique, notching out the shifted 20 KHz and then looking at the power in the residual after notch filtering. If there is no significant power, then there is no intruder. Conversely, if the residual power is above a set threshold, then it is assumed that an intruder is present.

The disadvantages of a Doppler scheme are as follows:

* The have to be on all of the time.
* They have high false alarm rates in situations that are active. If there are noticable air currents in a room, or anything that moves, they will be triggered. Also, they may be triggered by background noise.
* Averaging times will be large.

5. CANDIDATE PROCESSING ALGORITHMS.

Three algorithms will be discussed in varying degrees of detail. The three algorithms in decsending order of preference are:

* Comparing the envelope of the current pulse return with the average of previous ones (time history approach).

* Comparying the PSD of the current pulse return with the average of previous ones (frequency approach).

* Comparison of the cepstrum of the of the current pulse with the average of the previous ones (log frequency approach).

In all of the above, it is assumed that the single pulse type of signal is being employed. In most cases, the processing would be nearly the same for the other types of signals.

The reasoning behind the choice of these algorithms is simply that these are the types of detection structures that occur in classical detection theory. In general, optimal detection schemes tend to consist of a filter section which optimizes the signal power relative to the noise and an averaging scheme which computes the single parameter on which the detection choice is based. This averaging or matching can be done on the data, its autocorrelation or (equivalently) its power spectral density, depending on what is appropriate for the given situation.

If the statistical parameters of the signal and noise are known, preferably in closed form, it is often possible to obtain a formal solution to the problem, as is done in the standard text books on signal detection.

In this case there are too many unknown variables that make the finding of a formal optimal solution an impossibility: the transfer function characteristics of the magazines are unknown (and would have to be considered statistical data by themselves). Also, the nature of the noise is not simple: not only is the noise spectrum not flat, but it is changing with time.

In all of these approaches, it may be necessary to prefilter the data in order to reduce the background noise input. This problem is considered separately in subsection 5.4.

5.1. TIME HISTORY APPROACH.

In this procedure, the envelope is computed using the Hilbert transform implemented either with a filter or with the FFT. The main concept in the Hilbert transform is that of a 90 degree phase change: all of the frequency information is shifted by exactly 90 degrees in a Hilbert transform device. The Hilbert transformed time series is also a time series. It has the same frequency content and the PSDs of the original data and the Hilbert transform of the data are the same in the limit (there are some edge effects in any finite implementation). The Hilbert transform does not generate any new information, it just rearranges the data, and this rearrangement is very convenient: it makes the extraction of the envelope of the original time series very easy. For example, if x(i) has the form

$$x(i) = m(i)*\cos(2\pi aiT)$$

where a is in arbitrary frequency and m(i) is a relatively slowly changing modulation function, then y(i), the Hilbert transform of x(i) can be shown to be:

$$y(i) = m(i)*\sin(2\pi aiT)$$

Squaring and adding these yields:

$$x^2(i) + y^2(i) = m^2(i)*[\cos^2(2\pi aiT) + \sin^2(2\pi aiT)]$$

$$= m^2(i)$$

The sinusoidal term has disappeared, leaving the square of the modulation.

In its simplest form, a Hilbert transform filter could be set up as follows:

$$y(i) = (x(i+1) - x(i-1))/2$$

where x(i) is the input to the filter and y(i) is the output. That is, y(i) is the Hilbert transform of x(i). Taking the Fourier transform of this relationship yields:

$$Y(f) = (X(f)*\exp(j2\pi fT) - X(f)*\exp(-j2\pi fT))/2$$

$$= X(f)*(j\sin(2\pi fT))$$

The transfer function then is:

$$H(f) = j \sin(2\pi fT)$$

Thus, the phase angle is 90 degrees across the whole range, and the gain is unity af f = F/2, where it is supposed that the pulse would be set. As mentioned above, a Hilbert transformer is any operation that changes the phase by 90 degrees. There are many filters that will work: the above filter is a very simple form. More complicated ones can be obtained from the Parks-McClellan program for generating FIR filters (see section 5.4 for a reference on this subject). Or a similar effect can be had by Fourier transforming x(i), multiplying by j, and then taking the inverse transform.

No matter how it is done, the important thing is the 90 degree phase change. This is what permits the envelope to be computed.

Assume then that y(i) is the Hilbert transform of x(i) using the above simple version of a Hilbert transformer or a more complicated one if found necessary. Form the envelope z(i,u) of x(i) from

$$z(i,u) = \sqrt{x(i)**2 + y(i)**2}$$

The u index denotes that this is the uth pulse being processed. Each replication z(i,u) is assumed to have N points, so N lowpass IIR first order digital filters are set up:

$$\overline{z}(i,u) = (1 - \beta)*z(i,u) + \beta*\overline{z}(i,u-1), \quad i=0,\ldots N-1$$

The β parameter has to have the range $0 < \beta < 1$ in order for the filter to be usable and stable. The closer that β is to one, the more severe the action of the filter is. This is discussed in APPLIED TIME SERIES ANALYSIS, VOLUME I, where the transfer function and other properties of this filter are derived. Note that the filtering index here is u, not i.

The time history z(i,u) consists of N filtered envelope points. After the filters have run for a while, the result should stabilize and become a good estimate of the mean value of the envelope.

The test statistic rl at the lth pulse can be formed in several wasy. The simplest is the difference of the current input pulse and the averaged envelope :

$$r1 = \sum_{i=0}^{N-1} (z(i,u) - \overline{z}(i,u))$$

A second version, handy for analyzing the statisitical properties, is to normalize by dividing by the envelope:

$$r2 = \sum_{i=0}^{N-1} ((z(i,u) - \overline{z}(i,u)))/\overline{z}(i,u)$$

$$= \sum_{i=0}^{N-1} (z(i,u)/\overline{z}(i,u) - 1)$$

The test then consists of comparing the absolute value of these parameter with a prestored one.

5.2. Power Spectral Density Approach.

In the power spectra density approach, it is the power spectral density that is processed rather than the envelope of the data.

The power spectrum of the uth pulse, denoted S(k,u), would be computed in a manner similar to the definition in Section 3:

$$S(k,u) = \frac{|X(k,u)|^2}{P}$$

where X(k,u) is the Fourier transform of the uth pulse return.

This would be smoothed using a lowpass filter in a manner similar to the time history version:

$$\bar{S}(k,u) = (1 - \beta)*S(k,u) + \beta*\bar{S}(k,u-1), \quad k=0,...,N/2$$

Again, β is a fixed parameter that is close to but less than one, and $\bar{S}(k,u)$ is the smoothed version of the spectrum.

The simplest form of the test statistic would be to take differences and sum:

$$r3 = \sum_{k=0}^{N/2} [S(k,u) - \bar{S}(k,u-1)]^2$$

If the current spectrum S(k,u) and the previous smoothed spectrum $\bar{S}(k,u-1)$ are similar, then r3 will be very small. On the other hand, if one of them is different from the other, then the value computed for r3 will be large. If there is an intruder present, then presumably r3 will be greater than the prestored R3, and the alarm is then given.

The r3 parameter can be analyzed under the assumption that it is like a Chi-squared variable. As with all of these procedures, it would be necessary to have experimental data in order to be able to make conclusive statments about miss and false alarm rates. On the other hand, the arguments in Section 3.1 about the power spectrum lead to the conclusion that this will work best with a wide band function, and such a function will tend to admit more of the noise, and thus leading to degraded performance.

5.3. Cepstrum Approach.

The cepstrum in the main is the invention J. W. Tukey (the letters in the word are a rearrangement of the work spectrum). The recipe is as follows:

* Compute the PSD of the data.
* Compute the natural logarithm of the PSD.
* Compute the PSD of the logarithm.

As will be seen in the following example, this can be useful for for estimating echoes in data. In the final PSD, echoes tend to generate line spectra whose frequency corresponds to the delay time of the echo.

Assume that y(i) has the form

$$y(i) = x(i) + \beta * x(i - p)$$

where x(i) is an arbitrary time history, β is parameter whose magnitude is rather less than one and p is delay. In other words, y(i) has x(i) and a weak echo of x(i) occurring p*T seconds later.

Assume that $R_x(i)$ is the autocovariance of x(i). Then $R_y(i)$, the autocovariance of y(i) will be of the form:

$$R_y(i) = R_x(i) + \beta * R_x(i - p) + \beta * R_x(i + p)$$

That is, the echo shows up as two replications of the original autocovariance, one shifted to the left and one to the right, both by the original delay time.

Taking the Fourier transform of the autocovariance yields the power spectral density (see ATSA, page 317). In this case, this results in the following:

$$S_y(k) = S_x(k) + \beta * S_x(k) * \exp(-j2\pi kp/N) + \beta * S_x(k) * \exp(j2\pi kp/N)$$

$$= S_x(k) + \beta * S_x(k) * [2 * \beta * \cos(2\pi kp/N)]$$

$$= S_x(k) * [1 + 2 * \beta * \cos(2\pi kp/N)]$$

That is, the spectrum of y(i) is the spectrum of x(i) multiplied by

unity and a cosine function.

There are no approximations up to this point. In order to procede, the well known approximation

$$\ln(1 + x) = x$$

for x small will be employed.

When the natural logarithm of the above spectrum is taken, the result is:

$$\ln[S_y(k)] = \ln[S_x(k)] + \ln[1 + 2*\beta*\cos(2\pi kp/N)]$$

Applying the above approximation to the expression under the assumption that β is small yields:

$$= \ln[S_x(k)] + 2*\beta*\cos(2\pi kp/N)$$

Note that the delay term in this form appears as a simple cosine. Thus, when the spectrum is computed of this log spectrum, the delay term appears as a single line in the spectrum. After sorting out the units (not done here), the frequency at which the line occurs corresponds to the delay time of the echo.

The manner in which the cepstrum technique would be applied to this problem is simple: in the preceding section, the spectrum would be replaced with the cepstrum, and the averaging calculations would otherwise be done in an identical way.

Presumably, the various echoes of whatever signal was employed would appear as a series of lines in the cepstrum. If an intruder made a significant change in this line pattern, then the alarm would be sounded.

Some data should be taken in order to check the variability of the cepstrum, but the it is not considered very likely that this procedure will be employed in the final system. The reasons for this feeling are as follows:

* There are relatively few successful cepstrum applications known to the authors. In most other cases, autocovariance has worked better.

* The purpose of the cepstrum is to estimate delay. This is not a delay estimation problem.

* The taking of the logarithm is likely to amplify what-

ever noise is already to be found in the system.

* The authors know of no study on the variability of cepstra.

It should not be too difficult to compute some cepstra in order to get a better feeling for what they are like in this application.

5.4. Prefiltering.

It may be necessary to filter the data in some manner in order to reduce undesired noise from entering the processing. Such filtering could be accomplished using either analog or digital procedures. This subsection will look at the problem from the viewpoint of doing the filtering digitally.

The assumptions made here are that the signal is a burst of a single tone, nominally one millisecond, that the sampling rate is 80 K sps and that the center frequency of the tone is 20 KHz. That is, the spectrum of the tone is situated directly in the middle of the frequency band.

Figure 5.1 shows the transfer function in dB of a 31 tap FIR (finite impulse response) filter. This filter was generated using the Parks McClellan algorithm and program (McClellan, Parks and Rabiner, 'A Computer Program for Designing Optimum FIR Linear Phase Digital Filters,' IEEE TRANSACTIONS ON AUDIO AND ELECTROACOUSTICS, AU-21, 506-526, 1973). This program can be used to geneate lowpass, highpass, bandpass and notch filters, and Hilbert transformers and numerical differentiators.

The main lobe of the filter extends from about 16 KHz to 24 KHz. and the top of the stopband is about -33 dB down from the top of the passband. While the filter nominally has 31 taps, this is a special case: because of the symmetry of filter between 0 and the Nyquist folding frequency, every other tap is zero, so that in actual fact, the filter has only 15 weights total. As the filter was constrained to be symmetrical, it can be implemented with only 8 multilplies per input point plus somewhat more additions.

Thus, if one thousand point records are being processed, about eight thousand multiplies would be required to do the filtering. Probably 12 bit multiplications and 32 to 36 bit accumulations would be more than adequate. If only one burst per second is to be processed, this amount of calculation is relatively small, and may be within the capability of some of the 16 bit microprocessors to perform without an additional multiplier. The algorithm would have to be specified more exactly in order to be sure, but in any event, a single multiplier of almost any sort should be adequate is a microporcessor by itself is not.

For the cost of doubling the length of the filter, the main center lobe could be divided in half. Alternatively, the addition of a few more weights could be used to reduce the top of the stop band and improve filter performance in so far as rejection is concerned.

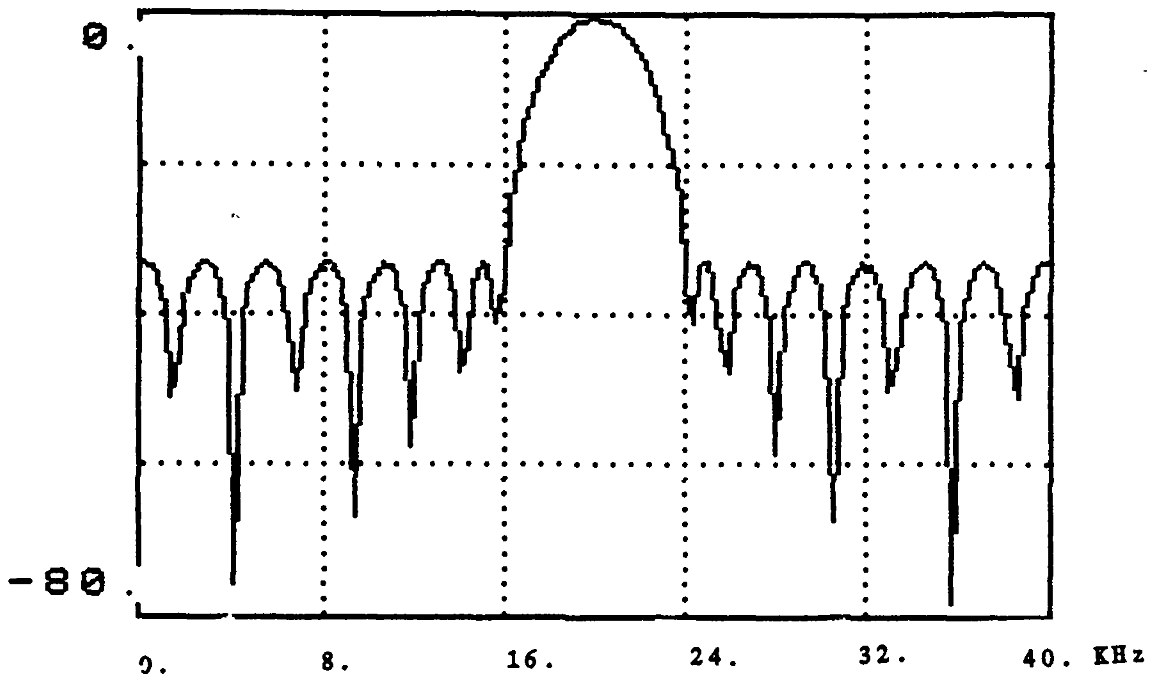


Figure 5.1. Filter transfer function.

6. STATISTICAL ANALYSIS AND FALSE ALARM RATES.

It is not possible to specify the false alarm rates and miss rates without additional information. This information will probably have to be obtained by taking actual measurements.

Nevertheless, it is possible to get considerable insight into the overall problem by looking at its statistical nature. In order to do this, a number of assumptions are made, the main one being that the time history approach of subsection 5.1 is the technique employed.

A somewhat modified form of the test statistic will be employed in this section. Suppose that the test statistic s is defined by

$$r = [m(z)/(\sqrt{N}\ s(z))] \sum_{i=0}^{N-1} ((z(i,u) - \overline{z}(i,u))/\overline{z}(i,u)$$

where

r = The test statistic.

m(z) = Overall mean value of the envelope.

N = Number of points being averaged.

s(z) = The overall standard deviation of the envelope.

z(i,u) = The uth occurance of the envelope.

$\overline{z}(i,u)$ = The uth occurance of the filtered and smoothed envelope.

The first assumption that is made is that the envelope averaging has reached steady state, and that the computed envelope is estimating the true envelope, in which case:

$$E[r] = 0$$

where the operation E[] means that we are taking expectations.

The next thing to be shown, under a number of strong assumptions, is that the variance of r is unity:

$$\text{Variance of } r = E[r**2] = s(r)$$

$$= [m(z)/(\sqrt{N}\ s(z))]**2\ E[\sum_{i=0}^{N-1} (z(i,u)/\overline{z}(i-u) - 1)**2]$$

$$= [m(z)/(\sqrt{N}\ s(z))]^{**}2 \sum_{i=0}^{N-1} [(s(z(i,u))^{**}2+m(z(i,u)^{**}2) / (\overline{z}(i,u)^{**}2 - 1]$$

At this point we make two strong assumptions:

$$m(z) = m(z(i,u))$$

$$s(z) = s(z(i,u))$$

They are not only strong, but likely incorrect. But they will allow us to model the process that is occurring. The assumptins are that the mean at each point in the echo is equal to the mean of the whole echo and similarly for the standard deviation (variance). This is a very rough approximation. Under these assumptions then

$$s(r)^{**}2 = [m(z)/(\sqrt{N}\ s(z))]^{**}2 \sum_{i=0}^{N-1} [s(z)/m(z)]^{**}2$$

$$= 1$$

What has been done is to set the test statistic up so that it has a zero mean and unit variance. By appealing to the central limit theorem, we can argue that it is Gaussian (almost any summation of a series of random variables is approximately Gaussian).

Standard tables of the Gaussian random variable show that:

$$1 - \int_{-4.86}^{4.86} p(r)\ dr = 1.157\ E-6$$

As mentioned at the end of Section 2, this is the probability level required to keep the false alarm rate to 3 per month when the test is run once every second. We can now combine all of this information in order to find out by how much the envelope has to change in order to have a detection. In other words, if we set the detection level at 4.86 and detect as follows

$r > 4.86$ or if $r < -4.86$	Intruder
$-4.86 < r < 4.86$	No intruder

then we will always meet the false alarm specification. The question remains, do we ever detect anything? (Also, what is s(z)?).

Next, for analysis purposes, assume that m(z) is one, so that the test statistic simplifies to

$$r = 1/(\sqrt{N}\ s(z)) \sum_{i=0}^{N-1} (z(i,u) - 1)$$

Assume further for a given intrusion, z(i,u) = 1+e for all i. This amounts to the signal rising by an amount e everywhere so that r is positive and

$$r = \sqrt{N}\ e/s(z)$$

In order to trip the threshold at the given false alarm rate, then

$$e = 4.86\ s(z)\ /\ \sqrt{N}$$

We would like to have e, the amount that the intrusion raises the level, to be as small as possible. There are two ways of accomplishing this:

* Make s(z) small, which amounts to minimizing the statistical variability of the envelope.

* Make N large be increasing the length of the record that is processed.

Except for the prefiltering and making a good choice of the test design, there is no way of decreasing s(z): whatever noise there is must be lived with. On the other hand, up to the limit of the reverberation time of the room, we may pick N, the number of points processed, as large as we like (we cannot make N indefinitely long: for any room, the echoes eventually die down, at which point the data is no longer usable). As an example, suppose N = 1024, so that its square root is 32. Then we would have e = 0.15 s(z) as the level the intrusion would have to achieve in order to trip the alarm level.

7. PROGRAMMING EXAMPLE.

This section will show how the pulse scheme could be implemented. The assumptions made in this example are follows:

* The 20 kHz, 1 millisecond pulse is employed as the signal, and an a rate of 80,000 samples per second is employed in the digitizing.

* The envelope detection scheme is used.

* For trial purposes, the coding is done in the FORTRAN language.

* The hardware to be used in the trial is a GenRad system having both an A/D and D/A.

* The code shown below in the main will be indicative rather than what would be finally employed in trials. It gathers statistics about the detection parameter r instead of doing actual detections.

```
      PROGRAM ALARM
      COMMON /HILBRT/ A,XX,YY,LAC
      COMMON /FILTER/ X,Y,B(16),M,LOC
      DIMENSION PULSE(101),RET(2001),AV(2001),VAR(2001)
C
C     SETUP.
C
C     NSIZE IS THE SIZE OF DATA USED IN THIS EXAMPLE.
C     THIS VALUE IS ARBITRARY AT THIS POINT.
      NSIZE=2001
C     FILTER WEIGHTS.  THESE WILL BE USED IN SMOOTHING
C     THE ENVELOPE OF THE RETURNED DATA.
      BETA=0.99
      ONMB=1.-BETA
C     NUMBER OF PULSES IN THE WARMUP.
      NWARM=100
C     NPULSE IS THE LENGTH OF PULSE.  IT IS 1 MSEC IN
C     LENGTH IF THE SAMPLING RATE IS 80 K SPS.
      NPULSE=80
C     ISTART IS THE  POINT AT WHICH CALCULATIONS BEGIN
C     IN THE RETURN PULSE.
      ISTART=200
C     SET UP THE A/D AND D/A.  THESE ARE CALLS TO SUBROUTINES
C     THAT DRIVE HARDWARE FOR A GEN RAD SYSTEM.
      CALL SETADS (Calling sequence omitted)
      CALL SETDAS (Calling sequence omitted)
C     DIGITAL FILTER WEIGHTS FOR THE BAND PASS FILTER.
C     A 31 TAP FILTER IS EMPLOYED.  AT IT IS SYMMETRIC
```

```
C       ABOUT THE CENTER, ONLY 16 WEIGHTS NEED TO BE
C       SPECIFIED.  THE ACTUAL WEIGHTS ARE NOT GIVEN HERE.
        M=31
        B(1)=
        .....
        B(16)=
C       INITIALIZE AV, THE AVERAGE ENVELOPE AND VAR, ITS
C       VARIANCE.
        DO 5 I=1,NSIZE
                AV(I)=0.
                VAR(I)=0.
5       CONTINUE
C
C       WARMUP.
C
        DO 20 N=1,NWARM
C       INITIALIZE THE FILTER AND THE HILBERT.
                LAC=0
                LOC=0
C               GENERATE THE PULSE.
                DO 10 I=1,NSIZE
                        PULSE=0.
                        IF (I.GT.NPULSE) GO TO 7
                        IT=I-4*(I/4)
                        IF (IT.EQ.1) PULSE=1.
                        IF (IT.EQ.3) PULSE=-1.
7                       RET(I)=PULSE
10              CONTINUE
C       SAMPLE IS A GEN RAD ROUTINE THAT IN THIS CASE
C       DOES BOTH THE A/D AND D/A.
                CALL SAMPLE (Parameters)
C       THE RETURN REPLACES THE ORIGINAL PULSE.
C       PROCESS THE RETURN.
                DO 15 I =1,NSIZE
                        X=RET(I)
C       BANDPASS FILTER THE DATA.
                        CALL BFILT
                        A=Y
C       COMPUTE THE HILBERT TRANSFORM.
                        CALL HILBRT
C       COMPUTE THE ENVELOPE.
                        ENV=XX*XX+YY*YY
C       COMPUTE THE AVERAGE OF THE ENVELOPE AND
C       ITS VARIANCE.
                        AV(I)=AV(I)+ENV
                        VAR(I)=VAR(I)+ENV*ENV
15              CONTINUE
20      CONTINUE
C       AVERAGE THE WARMUP COMPUTATIONS.
        FN=NWARM
```

```
      DO 25 I=1,NSIZE
              AVER=AV(I)/FN
              AV(I)=AVER
              VAR(I)=(VAR(I)-FN*AVER*AVER)/FN
25    CONTINUE
C     RAV IS THE AVERAGE OF THE THRESHOLD STATISTIC.
      RAV=0.
C     RVAR IS THE VARIANCE OF THE THRESHOLD STATISTIC.
      RVAR=0.
C     NPROC IS THE NUMBER OF PULSES TO PROCESS IN THIS RUN.
      NPROC=1000
C
C     MAIN LOOP FOR CALCULATIONS.
C
      DO 100 I=1,NPROC
C     SETUP.
              LOC=0
              LAC=0
C     R IS THE TEST PARAMETER.
              R=0.
C     GENERATE THE PULSE.
              DO 30 K=1,NSIZE
                      PULSE=0.
                      IF (K.GT.NPULSE) GO TO 27
                      IT=K-4*(K/4)
                      IF (IT.EQ.1) PULSE=1.
                      IF (IT.EQ.3) PULSE=-1.
27                    RET(K)=PULSE
30            CONTINUE
C     OUTPUT THE PULSE AND INPUT THE RETURN.
              CALL SAMPLE (Parameters)
C     COMPUTE R, THE TEST PARAMETER.
              DO 40 K=1,NSIZE
                      X=RET(K)
                      CALL BFILT
                      A=Y
                      CALL HLD
                      ENV=XX*XX+YY*YY
                      RET(K)=ENV
C     NOT ALL OF THE RETURN IS PROCESSED.
                      IF(K.LT.ISTART) GO TO 40
                      R=R+(ENV-AV(K))**2
40            CONTINUE
C
C     NORMALLY, AT THIS POINT R WOULD BE COMPARED WITH CAPR,
C     THE THRESHOLD OF DETECTION.  IN THIS EXAMPLE, WE ARE
C     ONLY GATHERING STATISTICS ON R, SO THE PROCESSING
C     CONTINUES.
C
C     AVERAGE THIS ENVELOPE IN WITH THE PREVIOUS ONES.
```

```
C
                DO 50 K=1,NSIZE
C       APPLY THE RECURSIVE LOWPASS FILTER TO SMOOTH THE
C       AVERAGED ENVELOPE OF THE RETURN.
                        AV(K)=BETA*AV(K)+ONMB*RET(K)
50              CONTINUE
C       COMPUTE AVERAGE OF THE STATISTICS.
                RAV=RAV+R
                RVAR=RVAR+R*R
100     CONTINUE
C
C       DONE.  COMPUTE THE FINAL VALUES FOR THE MEAN
C       AND VARIANVE OF R.
C
        FN=NPROC
        RAV=RAV/FN
        RVAR=(RVAR-FN*RAV*RAV)/(FN-1.)
C
C       At this point there would be some output.  The values for
C       RAV and RVAR and plots of AV, the averaged envelope would
C       be the most important.
C
        STOP
        END
        SUBROUTINE BFILT
C
C       SUBROUTINE FOR FIR FILTERING.
C
        COMMON /FILTER/X,Y,B(16),M,LOC
        DIMENSION BUFFER(16)
        IF (LOC.NE.0) GO TO 7
C       SET UP.
        DO 5 I=1,51
                BUFFER(I)=0.
5       CONTINUE
        LOC=1
        M2=M/2
C       DATA ENTRANCE.
7       BUFFER(LOC)=X
        LO=LOC-M2
        IF (LO.LT.1) LO=LO+M
        Y=BUFFER(LO)*B1
        DO 10 I=1,M2
                LL=LO-I
                IF (LL.LT.1) LL=LL+M
                LR=LO+I
                IF (LL.GT.M) LR=LR-M
                Y=Y+(BUFFER(LR)+BUFFER(LL))*B(I+1)
10      CONTINUE
        LOC=LOC+1
```

```
      IF (LOC.GT.M) LOC=1
20    RETURN
      END
      SUBROUTINE HLD
C
C     SUBROUTINE FOR THE HILBERT TRANSFORM FILTER
C
      DIMENSION B(19),W(5)
      COMMON/HILBRT/ A,X,Y,LOC
      DATA M/19/,M1/9/,M2/5/
      DATA W/0.62421036,0.17695171,0.07470435,0.02874678,0.007280592/
      IF (LOC.NE.0) GO TO 10
      DO 5 I=1,M
              B(I)=0.
5     CONTINUE
10    LOC=LOC+1
      IF (LOC.GT.M) LOC=1
      B(LOC)=A
      SUM=0.
      K=LOC-M1
      IF (K.LT.1) K=K+M
      DO 15 I=1,M2
              J1=K-2*I+1
              IF (J1.LT.1) J1=J1+M
              J2=K+2*I-1
              IF (J2.GT.M) J2=J2-M
              SUM=SUM+(B(J1)-B(J2))*W(I)
15    CONTINUE
      X=B(K)
      Y=SUM
      RETURN
      END
```

8. CONCLUSION AND RECCOMENDATIONS.

The authors of this report believe that an instrusion detection scheme based on signal processing techniques is feasible and has the potential for a low false alarm rate.

In order to verify this, it will be necessary to gather specific data and process it in a manner relevant to the various detection schemes.

The information that is needed in order to make a final design includes the statistical information about the test statistic r for the various candidate techniques. Specifically, accurate estimates of the mean, variance and the sample probability density function for each of the r parameters under the condition of no intruder being present are required. From this information it is possible to estimate false alarm rates for each of the methods. The false alarm rate is the key item used in evaluating a given procedure: the choosen system must have a low false alarm rate. If there are too many false alarms, the system users will simply turn it off. Depending on the users, probably no more than two or three false alarms per month will be tolerated.

In order to have useful results, the variability of the estimates of the sample mean and variance must be small. Thus, a large amount of data must be gathered and processed.

Fortunately, this is relatively easy to do. A prototype of the system using a minicomputer can be employed to test the algorithms and gather the required data. Over night runs using such a prototype can readily collect samples of size 25,000 to 100,000 estimates of r. This may seem large at first glance, but for this type of analysis such a sample size is the bare minimum.

The following plan is reccomended:

a. General considerations. A general purpose, programmable signal analyzer having both D/A and A/D capabilities, such as the GenRad 2510 should be employed in the collection of the data. In particular, the ISAP program and TSL languages provided by GenRad would be helpful in the initial stages of data collection, while the combination of FORTRAN and the GenRad TSALF routines would be required in later testing.

b. Data to analyze the performance of the pulse-envelope method, the pulse-PSD method and perhaps the pulse-cepstrum method should be collected.

c. The first step would be to collect the necessary equipment, generate the required software and assemble the

components and people at NSWC for the test period. TSA has a GenRad 2510 system which could be used for this purpose and could supply the software and people.

d. As noted above, what is needed here are the mean, variance and sample probability density function for the r test parameters for the various schemes being considered under the condition that there is no intruder. This would be obtained by setting the analysis system and software to run over night in a closed magazine like room at NSWC and generating and processing a large amount of data. One night a piece for each scheme should suffice (several weeks should be alloted...there will be breakage in the plan due to set up problems) to gather enough of this type of data after the required software and algorithms are debugged.

e. On the basis of the mean, variance and sample probability functions of the r terms, it will be possible to generate good estimates of the false alarm rates.

f. The next step would be to get some idea of the miss rate. This is much more difficult to do than gathering false alarm rate data. It may only be possible to verify that intrusion normally causes a very large change in r, making a miss highly unlikely.

g. After the completion of the pilot data gathering at NSWC, it would be advisable to analyze all of the data and issue an interim report.

h. After all concerned are in agreement about the conclusions of the interim report, further testing would be done by taking the GenRad system to sea and gathering data under actual conditions. It is assumed that NSWC and Navy personnel would perform this portion of the study.

i. After completion of the sea trials, the data would be reviewed and a final report with specific reccomendations on the the alarm system would be generated.

END

FILMED

2-85

DTIC